Development of an Optimized Fitting Routine for Comparing Theoretical Data with Experiments on Moisture Sensitive Beads for Carbon Dioxide Capture

by

Vinuta Chopra

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2016 by the
Graduate Supervisory Committee:

Klaus Lackner, Chair
Peter Fox
Allen Wright

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

Carbon dioxide (CO_2) is one of the most dangerous greenhouse gas. Its concentration in the atmosphere has increased to very high levels since the industrial revolution. This continues to be a threat due to increasing energy demands. 60% of the worlds global emissions come from automobiles and other such moving sources. Hence, to stay within safe limits, it is extremely important to curb current emissions and remove those which have already been emitted. Out of many available technologies, one such technology is the moisture swing based air capture technology that makes use of resin material that absorbs CO_2 when it is dry and releases it when it is wet. A mathematical model was developed to better understand the mechanism of this process. In order to validate this model, numerical simulation and experimentation was done. Once the mechanism was proved, it was seen that there are many factors and parameters that govern this process. Some of these do not have definite value. To find the best fit value for these parameters, an optimized fitting routine needs to be developed that can minimize the standard deviation of the error. This thesis looks into ways in which the optimization of parameters can be done and the possible future work by using substantial data.

DEDICATION

I dedicate my dissertation work to my family and friends. This is for you, Ma and Di. Thanks for sending me away and letting me chase my dreams. Thank you papa and amma for being my light and looking out for me. I'm indebted to you for life. I dedicate this work and give special thanks to my closest companion Piyush Awasthi and my best friends, old and new. Thank you for supporting me and uplifting me during the hard times.

ACKNOWLEDGMENTS

I sincerely thank my advisor Dr. Klaus Lackner for his continued guidance, support and encouragement during my masters and while writing this thesis. I also would like to thank Dr. Peter Fox & Mr. Allen Wright for being on my thesis supervisory committee. I would like to thank Piyush Awasthi, Sachin Grover and Prashant Sharma for being resourceful. I would like to thank Mr. Christophe Jospe and all my lab - mates Yang, Hannah Rose, Arvind and all members of Centre for Negative Carbon Emissions and at Arizona State University for making this journey so exciting and a great learning experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Rising global temperatures have increased different concerns from various fields in academia and industries. Climate change, involving rising sea levels, drought etc., are increasing the vulnerability of the environment. Extensive studies have shown that global warming is caused by greenhouse gasses, out of which, carbon dioxide (CO_2) plays a major role. Although methane (CH_4) is a more potent greenhouse gas, CO_2 is one of the major concerns because its emissions have risen exceptionally since the industrial revolution. The pre industrial atmospheric level of CO_2 was about 280 ppm. It has crossed 400 ppm in 2015 and is increasing at the rate of 2 ppm per year. According to the 5^{th} IPCC report, to stay under the targeted 2^0C rise in temperature, the maximum dangerous threshold for atmospheric CO_2 concentration is 450 ppm. This critical constraint calls for major and drastic measures to be taken to curb CO_2 emissions and help mitigate climate change.

The current technologies mainly focus on stationary sources, like power plants. They focus on removing the CO_2 from the source before they enter the atmosphere. The stacks of the power plants are fitted with CO_2 capture materials, which capture CO_2 at extremely high concentrations. Though, this is a pretty logical approach, it does not deal with all the CO_2 emissions. 60% of the global CO_2 emissions are from distributed and often mobile sources. Since, such emissions cannot be dealt with at the source, the CO_2 emitted must be captured from the atmosphere. Direct air capture (DAC) technologies are those which focus on capturing CO_2 from ambient air. One such technology is the moisture swing process.

The moisture swing process, developed by Lackner et al. uses the moisture in

the environment to capture CO_2 in the sorbent. It is capable of capturing CO_2 from the ambient air, which is at dilute concentrations. "Moisture swing," as the name suggests, involves using moisture to operate the process. The sorbent used in this process captures CO_2 when it is dry and releases it when it is wet. The entire cycle of a moisture swing can be schematically depicted as shown in figure 1.1. The heterogeneous sorbent, made of an anionic exchange resin, consists of quaternary ammonium ions embedded in polystyrene. It is made to be washed with hydroxide (OH^-) or carbonate (${CO_3}^{2-}$) solutions, thereby introducing these ions into the resin. The stoichiometry of the sorption of CO_2 on the resin shows that 1 mole of OH^- or ${CO_3}^{2-}$ ions in it can capture 1 mole of CO_2 and generate 1 mole of bicarbonate (${HCO_3}^-$) when exposed to dry air. As shown in figure 1.1 on the next page, for the wet resin without the CO_2 (Empty Wet), the ${CO_3}^{2-}$ ions behave as the counter ions for the quaternary ammonium ions. The presence of water stabilizes the carbonate ion. During drying, the reduction in the water content makes the carbonate ion less stable which results in splitting of the water molecules and reacting with the carbonate ions forming OH^- and ${HCO_3}^-$ ions (Empty Dry). At this state, the remaining OH^- ion has a strong affinity for CO_2 and reacts with it to form another bicarbonate ion (Full Dry). On wetting the resin again, the bicarbonate ions get completely hydrated (Full Wet). In this state, the equilibrium between carbonate and bicarbonate ions cause a high partial pressure of CO_2 allowing it to escape and go back to the original Empty Wet condition. This cycle repeats itself regenerating the sorbent. As, the moisture controls the flow of CO_2 in and out of the resin, this process is termed as a Moisture Swing.

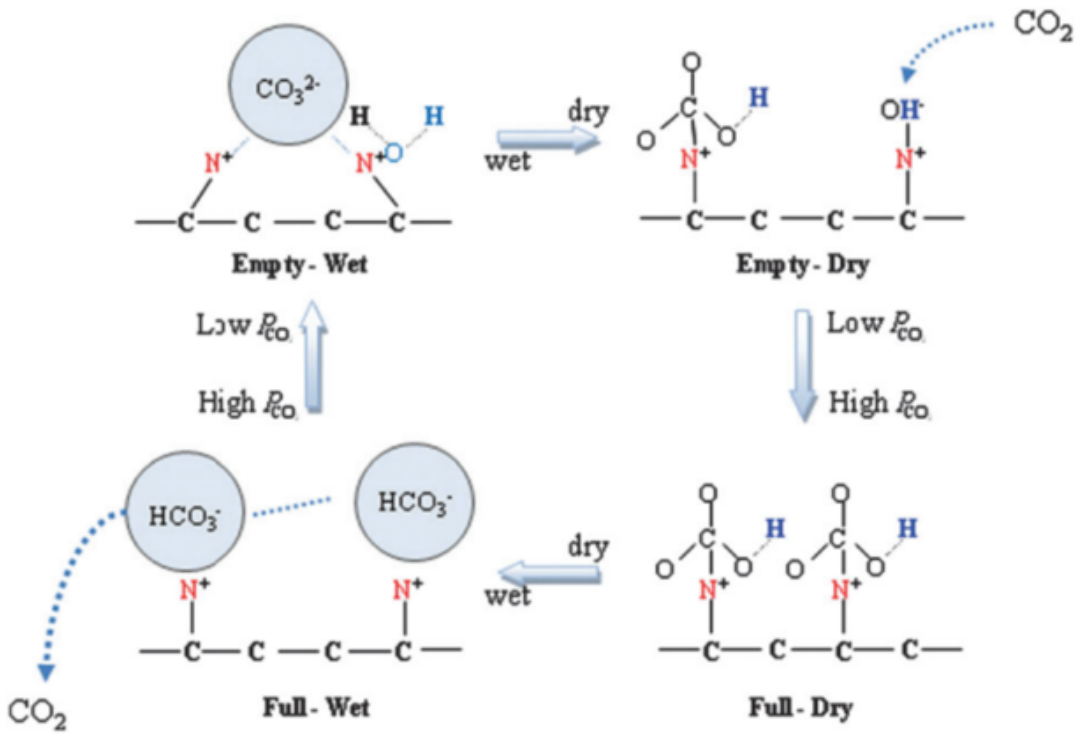

Figure 1.1: Reaction of CO_2 with the Sorbent during a Moisture Swing Cycle.

The resin is in the form of small spherical beads, each of about 0.6mm in diameter. Each bead may act as a CO_2 pump in the sense that the gas moves in and out of the bead based on the presence of moisture and partial pressure of CO_2 in the gas surrounding the bead. The behavior of these beads is shown by a diffusion model as shown in figure 1.2, the CO_2 in the atmosphere with lower partial pressure may enter the dry resin. It may diffuse from the drier region into the wet region in order to attain equilibrium with the water. At this state, since the partial pressure of CO_2 is higher within the resin, it diffuses out into the atmosphere.

Based on the above explained mechanisms, Isabelle Remy developed a model to describe the behavior of the moisture swing in the resin. It consisted of matter diffusion, thermodynamics, electrical fields and chemical equilibrium. The mesoscopic model developed by Isabelle would require the parameters: diffusion coefficients, elec-

trical mobilities, Henrys constants and chemical equilibrium constants. Based of on her work, Robin Abs improved this model to validate and calibrate it. He developed an experiment to gather the experimental data on the partial pressures of water and CO_2 within the system. He also ran simulations based on this model to find the best estimate of the theoretical data for these parameters. I will be furthering this project by analyzing the data collected by Robin and optimizing the fitting parameters using the conjugate gradient algorithm.

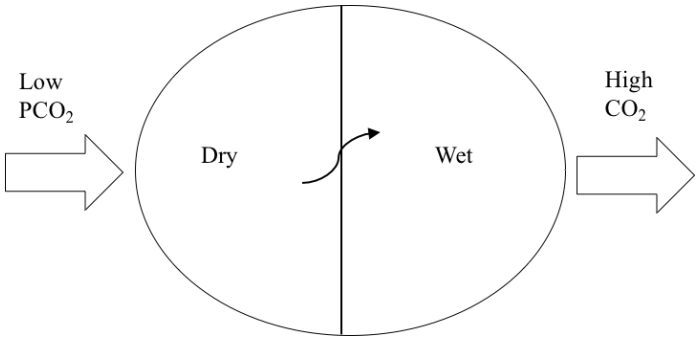

Figure 1.2: Diffusion Mechanism in the Bead of the Resin.

This report mainly consists of background, methodology, results and conclusion with future work. The background section of this report will summarize Isabelle and Robin's work. It will give details of the model proposed by Isabelle and the nature of the entire process. It will also show Robin's correction to the model, his experimental setup, the data he collected and what parameters are to be analyzed and optimized. The methodology consists of an introduction to the optimization algorithm. It will report on the differences between Steepest Descent and Conjugate Gradient methods of optimization and why I chose the latter over the former. The results section of the report will talk about the results of the optimization and the difference in the parameters values before and after optimization. The conclusion with future work section will give a brief summary of the entire report and discuss potential future work that can be done to further validate the system.

Chapter 2

BACKGROUND

This section of the report focuses on the work done by Isabelle and Robin. It gives a brief summary of the model developed by Isabelle, the nature of the process and the constraints and parameters that govern it. This section also talks about the improvements to Isabelles model by Robin, the experimentation and simulation developed by him to validate and calibrate the model.

2.1 Isabelle's Model

Isabelle's model describes the behavior of the sorbent. It is based on a hypothetical membrane made of a specific resin which has the properties to capture CO_2 when it is dry and releases it when it is wet. Isabelle worked on developing a physical model based on basic physical and chemical processes. She presents a theoretical model which is simplified to run computer simulations and observe the theoretical results. Her simulations were conducted with arbitrary parameters as she had no experimental source for the data.

Isabelle's report discusses two geometries of the resin. The first one is considered to be an infinite flat sheet dividing the space into two distinct regions: one containing dry air with constant moisture and the other containing wet air with constant moisture (Dirichlet condition). These two regions provide the boundary conditions for the two membrane surfaces of the resin. Consequently, water diffuses into the dry area from the wet area. This establishes a water flow from the wet side of the resin to the dry side. According to Isabelles model, another flow occurs. Theoretically, CO_2 will diffuse in the direction opposite to the water diffusion even though the CO_2

concentration in the gas phase on both sides are assumed to be equal. In the second geometry, she considers the resin to be a spherical and symmetrical bead as shown in figure 2.1 below. In this case, the moisture outside the bead is held constant with a flux condition, given by Neumann condition. An assumption is made that the flux at the center of the bead is null. Isabelle numerically showed that when a wet bead is placed in a dry environment, it becomes dry while capturing CO_2.

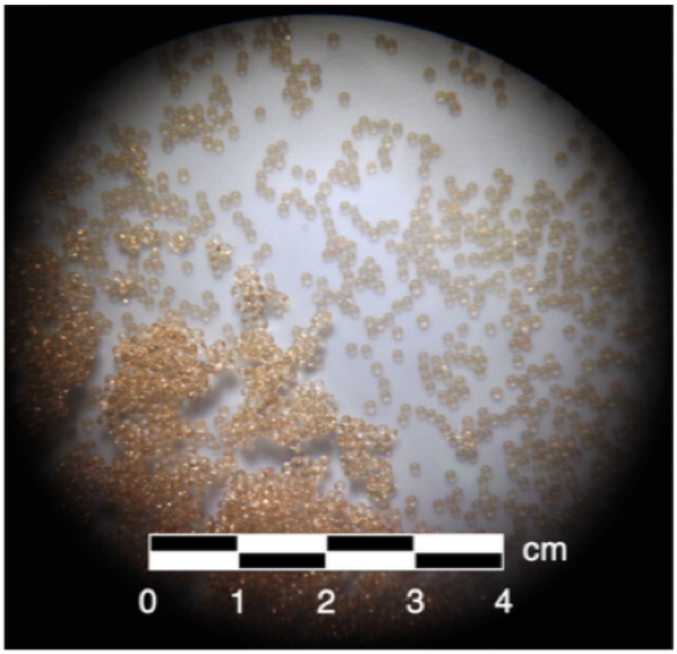

Figure 2.1: Microscopic View of the Beads in the Resin.

For furthering this project, Isabelle and Robin use the second case of geometry. The major hypothesis and approximations considered by Robin from Isabelle's report are as follows;

- The system is at constant temperature and pressure.
- The beads considered are all uniform, spherical, independent and indistinguishable.
- The cations inside the beads are quaternary ammonium ions and are immobile.

The anions (OH^-, CO_3^{2-} and HCO_3^-) are mobile and are responsible for the transport of the CO_2 in and out of the system.

- The global charge neutrality is always satisfied. That is,

$$2 * n_{CO_3^{2-}} + n_{HCO_3^-} + n_{OH^-} = n_+ \tag{2.1}$$

where n_k is the quantity of ionic species k per unit volume

- The null current condition is satisfied. That is,

$$2 * j_{CO_3^{2-}} + j_{HCO_3^-} + j_{OH^-} = 0 \tag{2.2}$$

where j is a flux inside system

- The entire system is divided into two parts: One on the surface of the resin and the other inside the resin, each governed by a different set of equations.

2.1.1 Equations Governing the Model

Isabelle proposed different equations that govern different parts of the resin on the surface, inside the resin and at the center. They are explained as follows;

- On the surface;
  - According to Henry's Law, the concentration of water on the surface of the beads can be calculated by,

$$P_{H_2O} = k_a * n_{H_2O} \tag{2.3}$$

  - The equations governing the equilibrium conditions for the reactions that occur on the surface are given by, $CO_2 + OH^- \overset{K_1}{\rightleftharpoons} HCO_3^-$

$$K_1 = \frac{n_{HCO_3^-}}{n_{OH^-} * P_{CO_2}} \tag{2.4}$$

$$HCO_3^- + OH^- \xrightleftharpoons{K_2} H_2O + CO_3^{2-}$$

$$K_2 = \frac{n_{H_2O} * n_{CO_3^{2-}}}{n_{OH^-} * n_{HCO_3^-}} = k_p * (n_{H_2O})^p \tag{2.5}$$

where k_p and p are constants

– According to electroneutrality;

$$2n_{CO_3^{2-}} + n_{HCO_3^-} + n_{OH^-} = n_+ \tag{2.6}$$

- Inside the resin;

The migration process of the moving species is divided into two components;

– The first one obeys Fick's Law and is expressed by the diffusion current $j_i^{diffusion}$. It deals with the species H_2O, OH^-, HCO_3^- and CO_3^{2-}. Here D_i is considered to be a constant.

$$j_i^{diffusion} = -D_i * \Delta_{ni} \tag{2.7}$$

– The second component, called drift current, applies to OH^-, HCO_3^- and CO_3^{2-}.

$$j_i^{drift} = n_i * \mu_i * F_i \tag{2.8}$$

Where μ_i is the ion's mobility and is considered a constant and

$$F_i = -q_i * \Delta\upsilon \tag{2.9}$$

– For the moving anions,

$$j_i = j_i^{diffusion} + j_i^{drift} \tag{2.10}$$

Isabelle divided the beads into nested shells with a thickness of r. She did mass balance over these shells at any time and got the equation,

$$n_i(r, t + \delta t) = n_i(r, t) - \delta t \left(\frac{\delta j_i}{\delta r}(r, t) + \frac{2}{r} j_i(r, t) \right) \pm \alpha(r, t) \tag{2.11}$$

- At the center; The flux at the center is given by,

$$\frac{\delta n_i}{\delta r}(r=0,t)=0, \forall t \tag{2.12}$$

At time t + δt,

$$n_i(r=0,t+\delta t)=n_i(r=0,t)-\delta t\frac{\delta j_i}{\delta r}(r=0,t)\pm\alpha(r=0,t) \tag{2.13}$$

2.1.2 Modification to the Model

According to Isabelles report, j_i of the moving ions was given by the following expression (assuming that $j_i = j_i^{diffusion} + c.\frac{\mu_i}{q_i}$)

$$j_i = j_i^{diffusion} + \frac{\mu_i}{q_i}e\frac{j_{OH^-}^{diffusion} + j_{HCO_3^-}^{diffusion} + 2j_{CO_3^{2-}}^{diffusion}}{\mu_{OH^-}+\mu_{HCO_3^-}+\mu_{CO_3^{2-}}} \tag{2.14}$$

For each anion specie, it is given by,

$$j_{OH^-} = j_{OH^-}^{diffusion} - \mu_{OH^-}\frac{j_{OH^-}^{diffusion} + j_{HCO_3^-}^{diffusion} + 2j_{CO_3^{2-}}^{diffusion}}{\mu_{OH^-}+\mu_{HCO_3^-}+\mu_{CO_3^{2-}}} \tag{2.15}$$

$$j_{HCO_3^-} = j_{HCO_3^-}^{diffusion} - \mu_{HCO_3^-}\frac{j_{OH^-}^{diffusion} + j_{HCO_3^-}^{diffusion} + 2j_{CO_3^{2-}}^{diffusion}}{\mu_{OH^-}+\mu_{HCO_3^-}+\mu_{CO_3^{2-}}} \tag{2.16}$$

$$j_{CO_3^{2-}} = j_{CO_3^{2-}}^{diffusion} - \frac{1}{2}\mu_{CO_3^{2-}}\frac{j_{OH^-}^{diffusion} + j_{HCO_3^-}^{diffusion} + 2j_{CO_3^{2-}}^{diffusion}}{\mu_{OH^-}+\mu_{HCO_3^-}+\mu_{CO_3^{2-}}} \tag{2.17}$$

$$j_{H_2O} = j_{H_2O}^{diffusion} \tag{2.18}$$

The correction for this is that, j_i should be $j_{H_2O} = j_{H_2O}^{diffusion} - \delta v.q_i.n_i.\mu_i$. Using this the final equation becomes

$$j_i = j_i^{diffusion} + q_i.n_i.\mu_i\frac{j_{OH^-}^{diffusion} + j_{HCO_3^-}^{diffusion} + 2j_{CO_3^{2-}}^{diffusion}}{e(n_{OH^-}\mu_{OH^-}+n_{HCO_3^-}\mu_{HCO_3^-}+4n_{CO_3^{2-}}\mu_{CO_3^{2-}})} \tag{2.19}$$

and for every component it will be,

$$j_{OH^-} = j_{OH^-}^{diffusion} - n_{OH^-}.\mu_{OH^-}\frac{j_{OH^-}^{diffusion} + j_{HCO_3^-}^{diffusion} + 2j_{CO_3^{2-}}^{diffusion}}{n_{OH^-}\mu_{OH^-}+n_{HCO_3^-}\mu_{HCO_3^-}+4n_{CO_3^{2-}}\mu_{CO_3^{2-}}} \tag{2.20}$$

$$j_{HCO_3^-} = j_{HCO_3^-}^{diffusion} - n_{HCO_3^-}.\mu_{HCO_3^-} \frac{j_{OH^-}^{diffusion} + j_{HCO_3^-}^{diffusion} + 2j_{CO_3^{2-}}^{diffusion}}{n_{OH^-}\mu_{OH^-} + n_{HCO_3^-}\mu_{HCO_3^-} + 4n_{CO_3^{2-}}\mu_{CO_3^{2-}}} \tag{2.21}$$

$$j_{CO_3^{2-}} = j_{CO_3^{2-}}^{diffusion} - 2n_{CO_3^{2-}}.\mu_{CO_3^{2-}} \frac{j_{OH^-}^{diffusion} + j_{HCO_3^-}^{diffusion} + 2j_{CO_3^{2-}}^{diffusion}}{n_{OH^-}\mu_{OH^-} + n_{HCO_3^-}\mu_{HCO_3^-} + 4n_{CO_3^{2-}}\mu_{CO_3^{2-}}} \tag{2.22}$$

$$j_{H_2O} = j_{H_2O}^{diffusion} \tag{2.23}$$

2.2 Robin's Model Calibration and Simulation

In order to simulate the model, Robin takes an analytical approach initially. This is done by solving the equations in Cartesian coordinates by non-dimensionalizing the parameters and the equations. Assuming that A = HCO_3^-, B = OH^-, C = CO_3^{2-} and D = H_2O, we get the following equations,

$$\frac{\delta n_A}{\delta t} = D_A \frac{\delta^2 n_A}{\delta x^2} - c\frac{\delta n_A}{\delta x} - f(n_A, n_B, n_C, n_D) \tag{2.24}$$

$$\frac{\delta n_B}{\delta t} = D_B \frac{\delta^2 n_B}{\delta x^2} - c\frac{\delta n_B}{\delta x} - f(n_A, n_B, n_C, n_D) \tag{2.25}$$

$$\frac{\delta n_C}{\delta t} = D_C \frac{\delta^2 n_C}{\delta x^2} - c\frac{\delta n_C}{\delta x} - f(n_A, n_B, n_C, n_D) \tag{2.26}$$

$$\frac{\delta n_D}{\delta t} = D_D \frac{\delta^2 n_D}{\delta x^2} + c\frac{\delta n_D}{\delta x} - f(n_A, n_B, n_C, n_D) \tag{2.27}$$

where $D_i \frac{\delta^2 n_i}{\delta x^2}$ is the diffusion term,

$-c\frac{\delta n_i}{\delta x}$ is the drift term

$\pm f(n_A, n_B, n_C, n_D)$ is the chemical term

These are the set of partial differential equations with non-linear terms. Here, we assume that c is constant although it generally contains electrical field and depends on n_i. To solve these equations in a temporary regime, we consider a finite domain where we use a separation of variables and solve the problem using boundary conditions, ending with the summation of all the solutions together.

To compute the theoretical model, Robin used a finite difference approximation to discretize the spatial and time derivatives. This was done by dividing the entire experimentation into finite grids as follows,

$$p_k = (k-1)\Delta p \tag{2.28}$$

$$\text{where k} = 1,\ 2,....,\text{K+1},\ \Delta p = \tfrac{1}{K}$$

$$T_m = (m-1)\Delta T \tag{2.29}$$

$$\text{where m} = 1,\ 2,....,\text{M+1},\ \Delta T = \tfrac{T_{tot}}{M}$$

After this, he also considered the Forward in Time Central in Space (FCTS) scheme to find the time first derivatives and space derivatives. The FCTS is conditionally stable, i.e., once we have chosen ΔT, Δp cannot be chosen arbitrarily or vice-versa. This is called Courant condition. For our case, which is the case of a simple diffusion, the stability condition is given by the Courant-Friedrichs-Lewy condition,

$$D\frac{\Delta T}{\Delta p^2} \leq \frac{1}{2} \tag{2.30}$$

In order to obtain stable solutions, we assume similar condition with water's diffusion coefficient, although our problem is not as simple.

2.2.1 P_{CO_2} Calculation

As the model is too complex to be solved analytically, Robin found another way to calibrate the parameters. This was done by comparing $P^{th}_{CO_2}$(from the model) and $P^{exp}_{CO_2}$(from the experiments) while imposing P_{H_2O}(t) and changing the parameters till the theoretical results and experimental results have the least discrepancy. Experimentally, the variables that can be modified are mainly P_{H_2O}, number of beads and volume of air. However,P_{H_2O} is the only variable that can be controlled by Peltier

element. Peltier element is a device in the experiment that controls the moisture in the sample chamber. In the non-dimensionalized model, the number of beads is irrelevant and working with the weight will be inaccurate in order to get a sinusoidal simulation.

The system we are dealing with is a closed system to measure the relevant partial pressures. This is also implemented in the model. Two different models are used to calculate the feedback from the beads on P_{CO_2}: One is the carbon conservation method and the other is the flux extrapolation method. The former method is more accurate and converges more quickly and the latter method requires a more precise grid which gives a pretty good evaluation of P_{CO_2}. Hence, we work with carbon conservation method. In this method we assume that at any time in our experiment, the total amount of carbon in the system has to be constant. Working on this with dimensionalized units and with first order approximations, we get,

$$n_{carbon}^{molar}(t) = \sum_{k=1}^{K} \left(\left(\frac{n_{HCO_3^-}(r_{k+1}) + n_{HCO_3^-}(r_k,t)}{2} + \frac{n_{CO_3^{2-}}(r_{k+1},t) + n_{CO_3^{2-}}(r_k,t)}{2} \right). \right. \tag{2.31}$$

$$\left. V_{shell_k} N_B \right) + n_{CO_2}(t).V_{air}$$

$$n_{carbon}^{molar}(t) = \sum_{k=1}^{K} \left(\left(\frac{n_{HCO_3^-}(r_{k+1}) + n_{HCO_3^-}(r_k,t)}{2} + \frac{n_{CO_3^{2-}}(r_{k+1},t) + n_{CO_3^{2-}}(r_k,t)}{2} \right). \right. \tag{2.32}$$

$$\left. 4\pi N_B r_k^2 \Delta r \right) + \frac{P_{CO_2}}{R\theta}(t).V_{air}$$

Let's define,

$$\lambda = \frac{V_{beads}}{V_{air}} \tag{2.33}$$

$$N_{carbon} = \frac{carbon^{molar}}{n_+.V_{air}} \tag{2.34}$$

$$N_{carbon}(t) = 3\lambda\Delta p \sum_{k=1}^{K} \left(\left(\frac{N_{HCO_3^-}(p_{k+1}) + N_{HCO_3^-}(p_k,t)}{2} + \frac{N_{CO_3^{2-}}(p_{k+1},t) + N_{CO_3^{2-}}(p_k,t)}{2} \right). \right. \tag{2.35}$$

$$\left. (p_k)^2 \right) + \pi_{CO_2}(T) N_{carbon}(T) = C^{te}, \forall T$$

Using the carbon conservation between T and T+1, the CO_2 partial pressure can be calculated as,

$$\pi_{CO_2}(T+1) = \pi_{CO_2}(T) - 3\lambda\Delta p \sum_{k=1}^{K} \left(\left(\frac{\Delta N_{HCO_3^-}(p_{k+1}) + \Delta N_{HCO_3^-}(p_k,t)}{2} + \right. \right. \tag{2.36}$$

$$\left. \left. \frac{\Delta N_{CO_3^{2-}}(p_{k+1},t) + \Delta N_{CO_3^{2-}}(p_k,t)}{2} \right).(p_k)^2 \right)$$

Where

$$\Delta N_{HCO_3^-}(p_k,T) = N_{HCO_3^-}(p_k,T+1) - N_{HCO_3^-}(p_k,T) \tag{2.37}$$

and

$$\Delta N_{CO_3^{2-}}(p_k,T) = N_{CO_3^{2-}}(p_k,T+1) - N_{CO_3^{2-}}(p_k,T) \tag{2.38}$$

2.2.2 Experimentation

The experimental setup consists of a closed loop system where P_{H_2O} is controlled and can be precisely measured along with P_{CO_2}. The experimentation results will be compared with the numerical simulation. The experimental setup as shown in the figure below essentially consists of a sample chamber, IRGA (Infrared Gas Analyzer), pump, Peltier element and an airflow controller. The resin is contained in the sample chamber and is arranged in such a way that the individual beads are all independent and do not touch each other. The IRGA is used to analyze the CO_2 concentration, H_2O concentration and pressure inside the system. The pump forces the air to circulate in the entire system. The Peltier element imposes the right value for P_{H_2O} in the system. It basically works on the Peltier effect (Thermoelectric effect) where when

heated, water evaporates and spreads in the whole system raising the P_{H_2O} and when cooled, the water condenses in the chamber, removing water from the system. The airflow controller controls the amount of air that goes into the chamber. The Peltier element if far enough from the measurement chamber so that the temperature in the gas returns to the normal ambient temperature before it comes in contact with the resin beads.

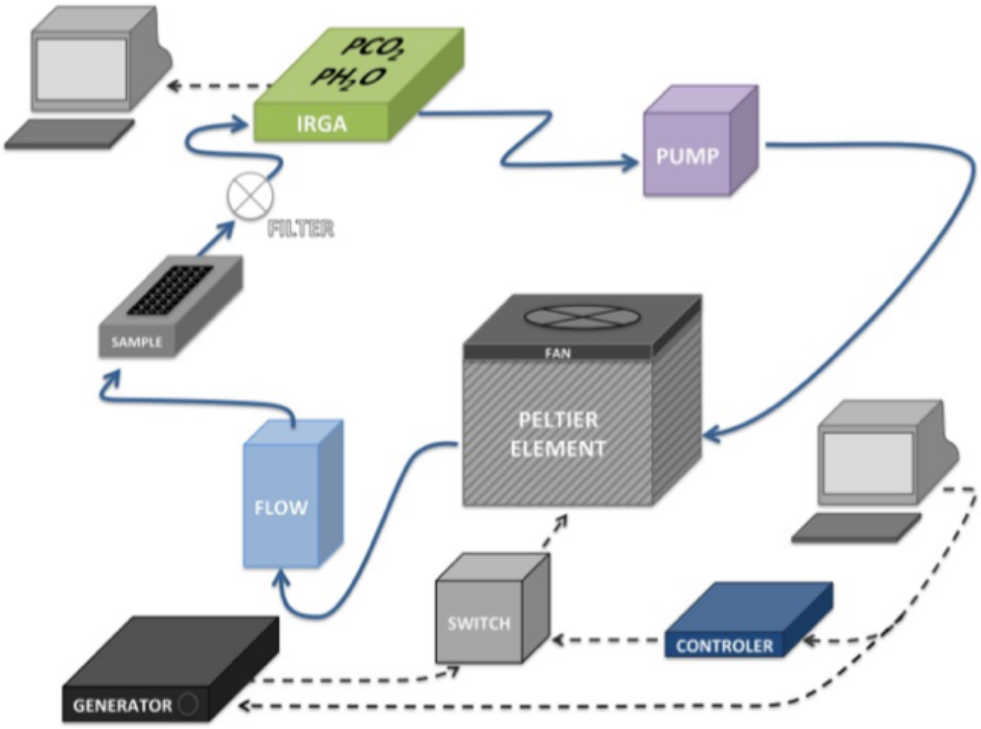

Figure 2.2: Experimental Setup.

To help monitor the system and control the different parameters needed for the experimentation, LabVIEW was used. LabVIEW is a software tool for collecting data from sensors and controlling hardware and actuators to adjust the behavior of a system. The operation of the various individual components was governed using this LabVIEW. The main drawback of this system is the CO_2 leakage issues. These leakages were found to be in the pump and airflow controller. The problem from

the pump comes from an improper sealing of the pumping chamber. Hence, all the parts to the pump need to be aligned and sealed in order to avoid leakage. In the case of the airflow controller, we simply removed it. The airflow controller does not have a real impact on P_{H_2O} measurements and is irrelevant in the model as well. Since, the experimentation needs to be 100% leak proof in order to get accurate P_{CO_2} measurements, we just removed the airflow controller.

Once the system is made leak proof, the next step is to control the P_{H_2O} in the sample chamber. This is done by using a Proportional-Integral-Derivative (PID) controller. It basically works by measuring the evolution of the error between $P_{H_2O}^{measured}$ and $P_{H_2O}^{setpoint}$. It then applies a correction to the current that drives the temperature in the Peltier element. This correction will be proportional to the error, its history and to the evolution of this error. Tuning is done by finding the proportional, integral and derivative terms. The fitting depends on the relative importance of each term. The calibration is done by first measuring the amplitude response of P_{H_2O} to a step of current (Gain), the delay and its characteristic time constant. The following two graphs show the steps of the current and the corresponding response of P_{H_2O}.

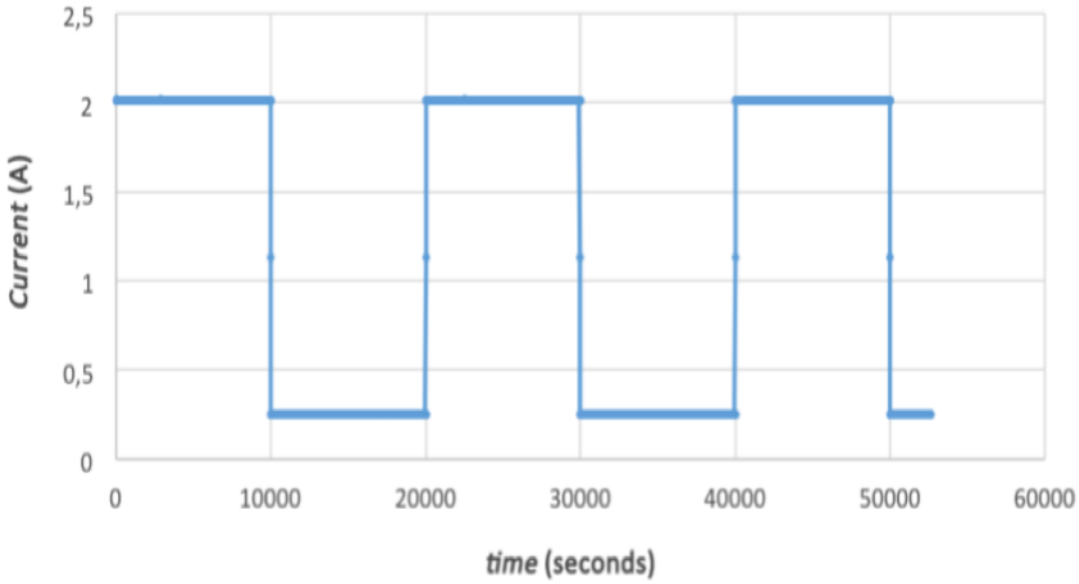

Figure 2.3: Steps in Current. Gain = -8.64, Delay = 12 s, Tchar = 218 s

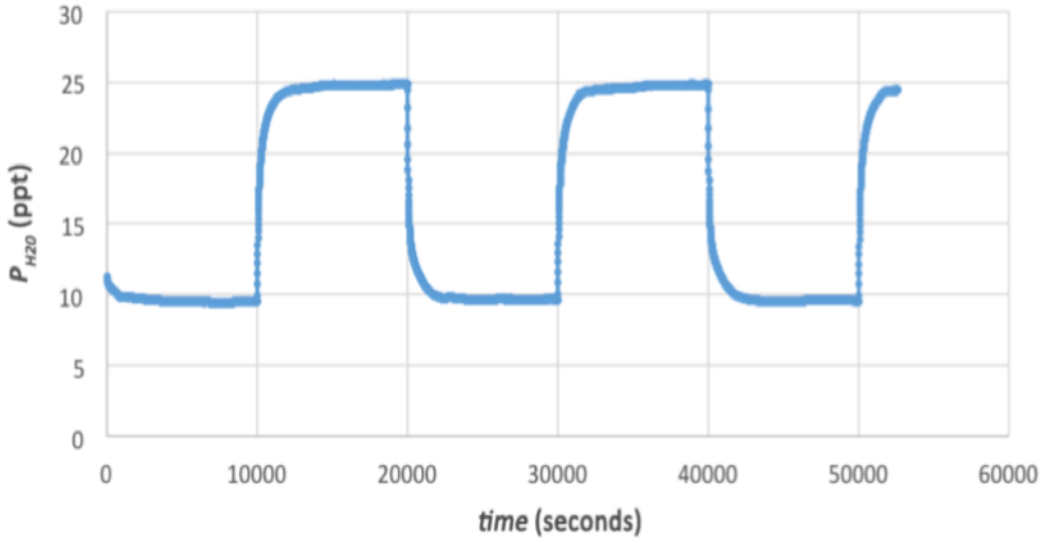

Figure 2.4: Response.

Using this data, Robin found the three parameters proportional gain (K_c), integral time constant (T_i) and derivative time constant (T_D) using Internal Control Method (IMC). After performing a correction on the formulas obtained from the IMC method, he gets the following values,

$$K_c = \frac{0.769}{Gain} * \frac{T_{char}}{Delay} = \text{- 1.62(-1.101 after correction)}$$

$$T_i = T_char = 218\text{s (229 after correction)}$$

$$T_D = \frac{1}{2} \text{ Delay} = 6\text{s (3.79s after correction)}$$

The corrective currents are directly related to these intermediate parameters and are implemented in LabVIEW using the following formulas,

$$\mu(k) = \mu_p(k) + \mu_I(k) + \mu_D(k)$$

$$\text{where } \mu_p(k) = (K_c.e(k)) \text{ is the proportional correction}$$

$$\mu_I(k) = \mu_I(k-1) + \frac{K_c}{T_i}.\frac{e(k)+e(k-1)}{2}\Delta t \text{ is the integrative correction}$$

$$\mu_D(k) = -K_c\frac{T_D}{\Delta t}(P_{H_2O}^{measured}(k) - P_{H_2O}^{measured}(k-1)) \text{ is the derivative correction}$$

$$\text{k = index of sampled signal at time k.t}$$

$$\Delta\text{t = sampling time of controller}$$

$$e(k) = P_{H_2O}^{setpoint}(k) - P_{H_2O}^{measured}(k-1)$$

Other issues faced during experimentation are outside temperature and pressure changes. The temperature changes affect the beads chamber and pump as they are in contact with the working bench. This is overcome by isolating them from the bench using foamed polystyrene. The pressure change is much more complex to deal with. P_{CO_2} is very sensitive to variations in P_{H_2O}. In our case, a 0.51% global pressure variation would cause a 25.7% variation in P_{CO_2}. The P_{CO_2} variations are measured while imposing constant P_{H_2O}. This causes a 27.2% P_{CO_2} variation caused by a 0.83% total pressure variation. Since the only pressure readings we get are from the IRGA, it cannot account for the direct measures of pressures inside the chamber. Hence, it is safe to say that there is a pressure drop from the outlet of the pump to its inlet. We assume that concentrations are uniform in the entire system. After significantly reducing the pressure variations, the pressure drop across the system is accounted for when evaluating P_{H_2O} using the following formulas,

$$P_{chamber} = P_{IRGA} + P_{drop}\ (P_{H_2O}) = P_{H_2O,chamber} = C_{H_2O} * (P_{IRGA} + P_{drop})$$

Using $P_{drop} = 1kPa$, the pressure correlation is significantly reduced and reaches an acceptable stage where the measured variation is 4.36% P_{CO_2} for a 0.93% IRGA pressure variation. Ideally, the best way to get rid of this issue is to directly measure the $P_{chamber}$.

2.2.3 Implementation and Parameters Analysis

The model is governed by 8 non-dimensionalized parameters and 3 dimensional parameters. The non-dimensionalized parameters are $D_{OH^-}, D_{HCO_3^-}, D_{CO_3^{2-}}, K_1^{'}, k_a^{'}, \lambda, k_p^{'}$, k_c. The dimensional parameters are τ (in terms of D_{H_2O}), n+ and R. The dimensional parameters are known, R = 600 m which is given by the supplier and n+ = 2095 mol m^{-3}. However, on titration, n+ was found to be 1620 mol m^{-3} which is very close to the supplier's value. The probable reason for the lower value could be that the beads were already absorbing CO_2 and releasing OH^-. The characteristic time, τ and D_{H_2O} completely rely on the experiment. This is done by accounting for the weight of the beads and the amount of water it absorbs. When P_{H_2O} is increased, the beads absorb the water before the interaction with CO_2 begins. After experimentation, it was found that the best fitting value for $D_{H_2O} = 4.2 * 10^{-11} m^2 s^{-1}$ with = 15.9 mins. It is given by the relation,

$$D = \frac{(\frac{R}{3})^2}{\tau} \tag{2.39}$$

For the parameters to be the best fit, Robin conducted the experiments for two time frames: one long and one short. Then, the fitting of the parameters was done by hand. The values of these are as shown below,

$$C_{H_2O} = 1,\ C_{HO^-} = 8.10^{-2},\ C_{CH_3O^-} = 1,2.10^{-1},\ C_{CO_3^{2-}} = 1,5.10^{-1},\ K_1^{'} = 4.10^9,$$
$$k_a^{'} = 10^{-5},\ \lambda = 4.10^{-3},\ k_p^{'} = 5.10^{-12} \text{ and } p = 9,5$$

In order to systematically compare the experimental and theoretical values, we compare the P_{CO_2} Fourier spectrum from the model and from the experiment for various frequencies. This is done to quantitatively evaluate the accuracy of the model among the whole frequency spectrum. During experimentation, we excite the system with a single frequency at four different times and equilibrate between them. The following two graphs show the frequencies occurred during experimentation and simulation respectively.

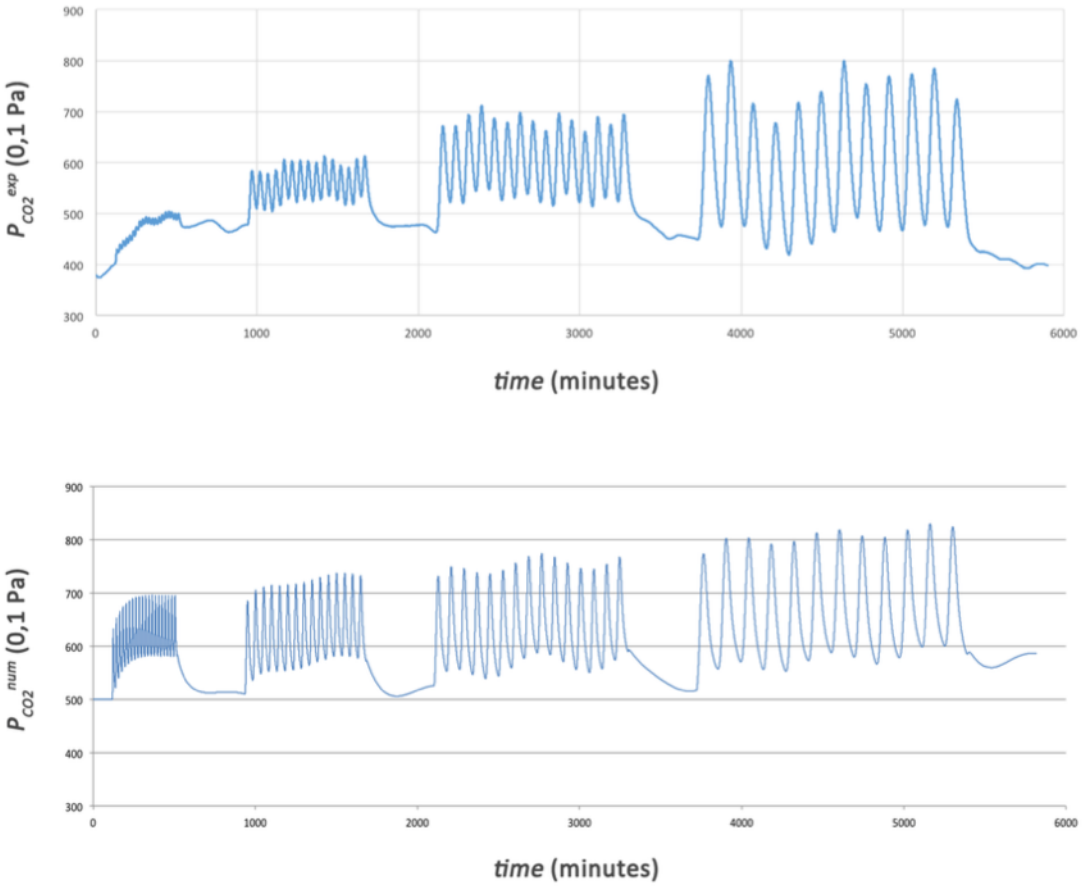

Figure 2.5: Experimental and Modeled P_{CO_2} Frequency Response.

Using Fast Fourier Transform and comparing the two graphs at relative frequencies and phase, we can determine the accuracy of the model. Though the frequency graph shows that our system is largely nonlinear, the phase graph shows closer results. They

are as shown in Figure 2.6 and Figure 2.7 as shown below.

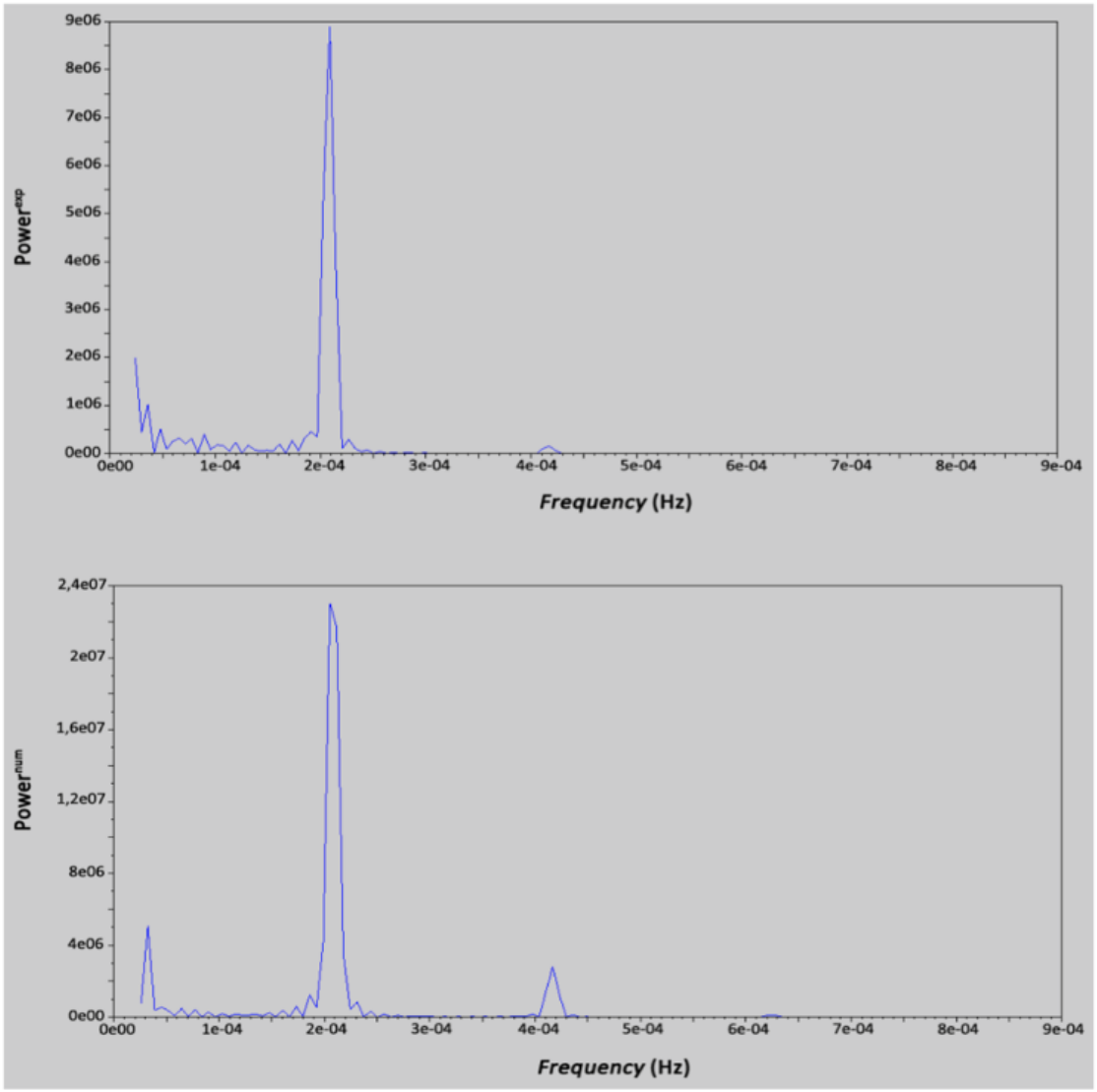

Figure 2.6: Experimental and Simulated P_{CO_2} Frequency Spectrum.

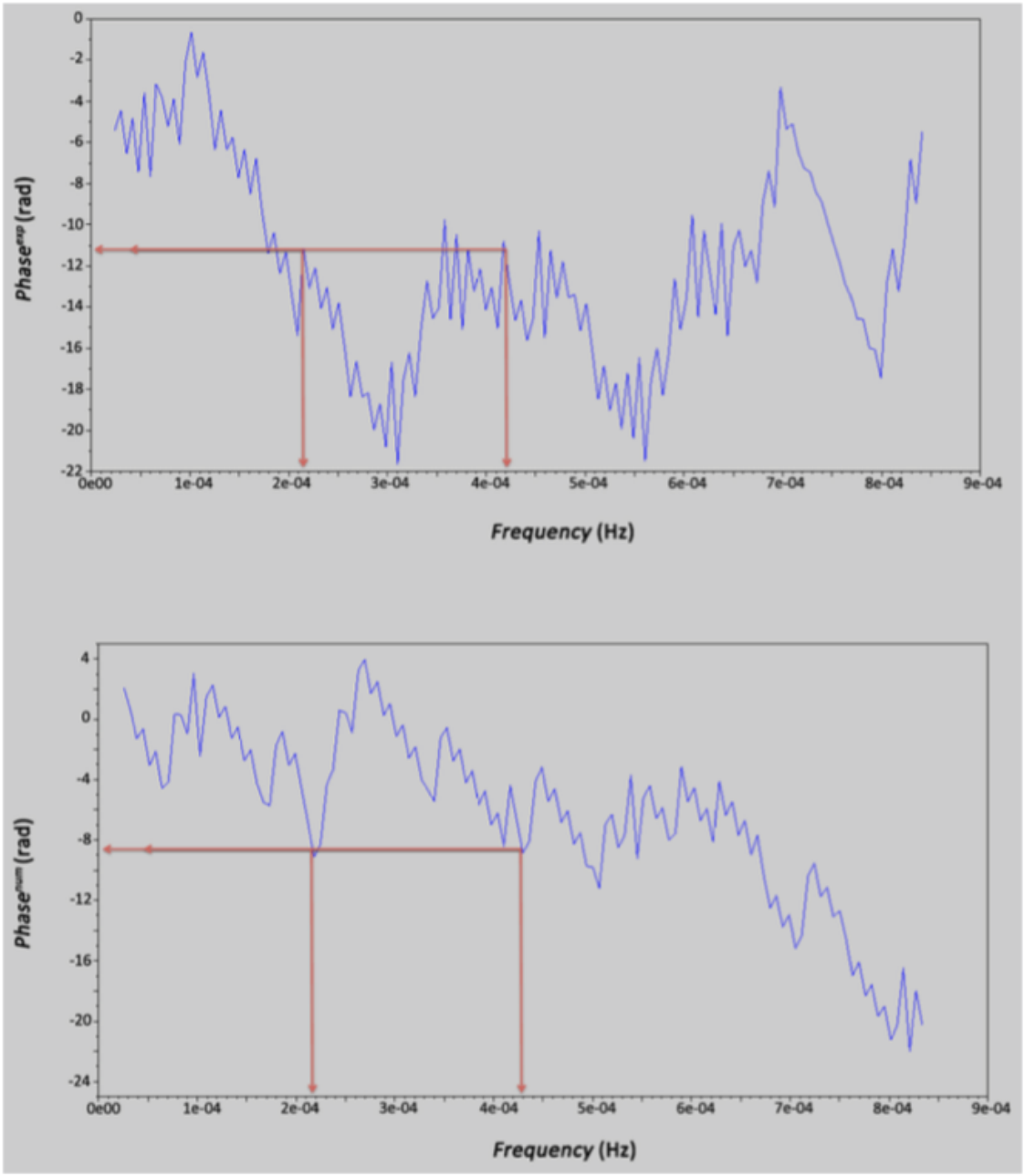

Figure 2.7: Experimental and Simulated P_{CO_2} Phase Spectrum.

After this, he input all the values obtained and found that a CO_2 gradient is created by the membrane because of the opposite H_2O gradient. The second conclusion is that H_2O diffuses faster than the ${CO_3}^{2-}$ ions. Finally, it was found that the carbon diffusion profile is more curved than anticipated in Isabelle's model.

Chapter 3

DATA ANALYSIS AND OPTIMIZATION METHODOLOGY

From the results of Robin's report, it is quite evident that the gradient in P_{H_2O} causes the flow of CO_2 in and out of the resin beads. The closeness of the numerical simulation with the theoretical model validates Isabelle's theory of the mechanism of moisture swing in the resin beads. This was done by using different non-dimensionalized parameters and some dimensional parameters. After analyzing the data and parameters used by Robin, we saw that he had worked by hand to guess the the values of certain parameters (the non-dimensionalized parameters). In this section, we address this issue in the numerical simulation by initially using arbitrary values for these parameters and finding the best fit using optimization. Since, we are using a non-linear system, we perform Fast Fourier transform for different values of frequencies for P_{H_2O}, obtain the solutions to the parameters and then use the optimization algorithms on it. The following section provides a detailed description of the process.

3.1 Initial Work

To continue from where Robin left off, it is important to get different experimental data and numerical simulation data for P_{CO_2} and P_{H_2O}. For this, it is important to set up the experimentation and get it working. However, when we started working on getting the experimental values, the system had a high leakage in the sample chamber. The water vapor would get condensed which would result in inaccurate values. To deal with this issue, we had to open all the components of the sample chamber to let it dry and reseal it to conduct the experiment. This was done in cycles which was extremely time consuming. However, the leakage issue was fixed and the experiment

is now ready to be resumed.

In the meanwhile, to get the numerical simulation results, I have to run Robin's code at different frequencies. At the first attempt, for 15000 frequency input, I was getting results in the range of 10^{19} while the expectation was around 400-500 ppm range. This high magnitude of difference was a major concern. On further analysis, I realized that the problem was in the semantics of code execution and data exportation. Once this was fixed, the code was ready for execution. The next steps were to get data for P_{H_2O} and P_{CO_2} using different frequencies. Four runs of different frequencies would provide one set of solutions for the eight parameters. Hence, a bunch of runs must be done before we get enough amount of data which can be optimized. This is the broader picture of the idea behind this thesis.

3.2 Fourier Analysis

For an accurate manipulation of the data that we have gathered, we need an efficient computational tool for accomplishing this. Fourier transform is one such method. It is mainly used to define physical processes in time domain and frequency domain. It is a linear operation where the sum of two functions is equal to the sum of the transforms. Based on the kind of data that we need to manipulate, different forms of Fourier transform may be used. Our data happens to be what is called discretely sampled data. In such cases, the function is sampled at periodic intervals of time. Using this, if we need to analyze 'N' points of data, we use Fast Fourier Transform or FFT.

To use FFT for our data, we first need to run Robin's code. This will give us the sinusoidal waves of P_{H_2O} and P_{CO_2} in the temporal domain. We can now use an FFT to transform the data to the frequency domain. For a linear system, stimulated by a

single driver, $\Delta P_{H_2O} = A\sin(\omega t)$. This is represented by the equation;

$$\Delta P_{CO_2} = A\sin(\omega t) + B\cos(\omega t) \tag{3.1}$$

where ωt is the phase of the signal

The values of A and B depends on the peaks obtained from FFT. If the output of FFT has noise, it is essential to smoothen them out using a Gaussian. This is usually available as an in built function in almost all softwares. It is given by the formula;

$$w(n) = e^{((-1/2)(\alpha.\frac{n}{(N-1)/2})^2)} = e^{(-n^2/2\sigma^2)} \tag{3.2}$$

When looking at the FFT graph, if the zero frequency is very large, it means that we have an average non - zero value. This must be subtracted before you take the FFT. Also if the width of the peaks seems to be about the same, it is essential to make it standard. This is when we can use Gaussian by multiplying the signal with a Gaussian about one third the width of the time window. This will smear about the peaks in a well defined manner and we can then estimate the strength at the two peaks at positive and negative frequencies as an integral over a narrow window. The Fourier transform of a Gaussian is another Gaussian with the inverse width. The Fourier Transform of a product of functions, is the convolution of the two functions in Fourier Space. Hence, if we multiply the function in the temporal domain with a wide Gaussian, we convolve its Fourier Transform with a narrow Gaussian. This filters out noise, because it is quite possible that the width of our window and the period are not commensurate and that our important signal is smeared in a random way over a few frequencies. If on the other hand we convolve it, then we smear it out in controlled fashion, where all we care about is that the width of the result is a little more than a single frequency. Then we can estimate the overall intensity from the integral over a window of this width.

3.3 Optimization Algorithms

After analyzing the data we have, the Conjugate gradient optimization method seems to be the best approach. Essentially, the conjugate gradient algorithm finds the minimum of a function that depends on a number of variables. It is an improved version of the steepest descent method which relatively takes much more calls to perform a given task. Both of these methods are discussed here.

3.3.1 Steepest Descent

The Steepest Descent method is a minimization algorithm which is generally used to optimize a sparse range of data. Conjugate Gradient is an improved version of Steepest Descent. Both these optimization algorithms are iterative methods used to solve a large system of linear equations. It is essentially used in a system of the form

$$Ax = b \tag{3.3}$$

where A = known, square, symmetric and positive - definite (or non-definite) matrix

x = unknown vector

b = known vector

In the method of steepest descent, let us start at an arbitrary point (x_0) and slide down till we reach the bottom of the paraboloid. For this a number of steps must be taken till we reach the desired result. Let us assume these series of steps to be x_1, x_2, x_3 and so on. In the steepest descent method, a step is taken in the direction where the given function 'f' decreases most quickly. For this, another routine called Line Search or Line Minimization is implemented to find the minimum point in the direction which is a line. This line search subroutine is implemented many times till we reach the final point. The term error 'e' is a vector that indicates how far we are

in the solution. The residual 'r' indicates how far we are from the correct value. The value of the residual indicates the direction of the steepest descent. The equations that govern these values are shown in the algorithm.

For a better understanding of this concept, consider the figure below. After choosing an arbitrary point x_0, the line minimization subroutine finds the minimum on a line towards the x axis. Once this is done, it is called again to find a minimum on another line which is at right angles to the previous line. On implementing this iteratively, we can see in the figure that it finally reaches the desired value of x.

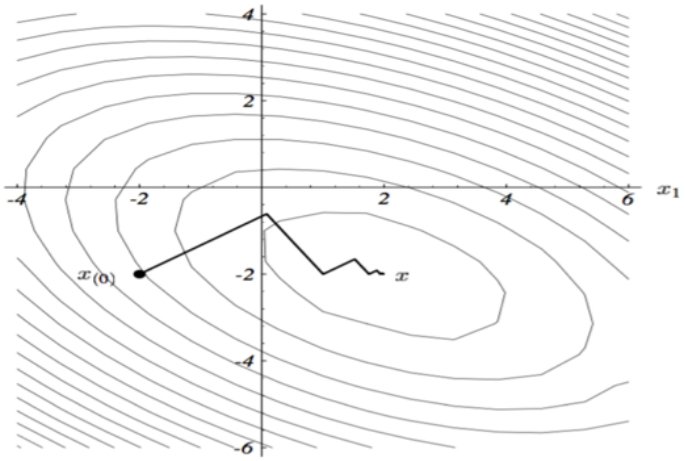

Figure 3.1: Steepest Descent Method.

The algorithm of the line search/line minimization (linmin) is as shown below. Here, the input are the vectors and a function a and the output is the minimum on the line produced.

> `linmin`: Given as input the vectors **P** and **n**, and the function f, find the scalar λ that minimizes $f(\mathbf{P} + \lambda\mathbf{n})$. Replace **P** by $\mathbf{P} + \lambda\mathbf{n}$. Replace **n** by $\lambda\mathbf{n}$. Done.

Figure 3.2: Terminologies.

The basic algorithm for the Steepest Descent is as shown below. Here, the inputs are A, b, the starting value x, the maximum number of iterations i_{max} and an error

tolerance ϵ ¡ 1. The initial iteration is set to zero. The algorithm terminates when i_{max} is exceeded. The fast recursive formula for the residual is used usually. But, it is exactly calculated once every 50 iterations (arbitrary). If n is large the number of iterations will be $\sqrt{n}$.

$i \Leftarrow 0$
$r \Leftarrow b - Ax$
$\delta \Leftarrow r^T r$
$\delta_0 \Leftarrow \delta$
While $i < i_{max}$ **and** $\delta > \varepsilon^2 \delta_0$ **do**
    $q \Leftarrow Ar$
    $\alpha \Leftarrow \frac{\delta}{r^T q}$
    $x \Leftarrow x + \alpha r$
    If i **is divisible by 50**
        $r \Leftarrow b - Ax$
    else
        $r \Leftarrow r - \alpha q$
    $\delta \Leftarrow r^T r$
    $i \Leftarrow i + 1$

Figure 3.3: Steepest Descent Algorithm.

The problem with the Steepest Descent method is that it performs many small steps in finding the minimized value even if the set of linear equations are in the perfect quadratic form. This is because the new gradient at the minimum point of the line minimization is perpendicular to the previously traversed direction. Hence, in every iteration it becomes necessary to take a right angled turn which does not always reach the desired minimum value. Therefore, to overcome the inefficiency in this algorithm, we use the conjugate gradient algorithm. Essentially, the conjugate

gradient algorithm does the same minimization as is done in the steepest descent method but, in many fewer steps. It works in a way that it does not go down the gradient but, traverses a path which is conjugate to it. It is explained more in detail in the following section.

3.3.2 Conjugate Gradient

To best understand conjugate gradient, let us consider an example. Consider, an N-dimensional point P, the value of a function f(P) and its gradient ∇f. The function is given by the quadratic equation,

$$f(x) \approx c - b \cdot x + \frac{1}{2} x \cdot A \cdot x \tag{3.4}$$

Here, the number of unknown parameters is the same as the free parameters in A and b, which is of the order N^2. Changing any one of these parameters can move the location to minimum. In this method, we want the path traversed by the linmin algorithm to be in conjugate to its previous direction. The conjugate gradient algorithm is a method to solve linear algebraic equations by minimizing a quadratic form or it can be applied to a minimizing function approximated by a quadratic form.

Let us start with an arbitrary value g_0. Assuming $h_0 = g_0$, the conjugate gradient method develops 2 sequences of vectors from the recurrence,

$$g_{i+1} = g_i - \lambda A.h_i \tag{3.5}$$

$$h_{i+1} = g_{i+1} - \gamma_i h_i \tag{3.6}$$

where i = 0,1,2,....

These vectors satisfy the orthogonality and conjugacy conditions, i.e, $g_i \cdot g_j = 0$, $h_i \cdot A \cdot h_j = 0$ and $g_i \cdot h_j = 0$ where j ¡ i. The scalars used in these equations are given

by,

$$\lambda_i = \frac{g_i \cdot h_i}{h_i \cdot A \cdot h_i} \tag{3.7}$$

$$\gamma_i = \frac{g_{i+1} \cdot g_{i+1}}{g_i \cdot g_i} \tag{3.8}$$

When the Hessian matrix 'A' is known, we could use these equations to successively find conjugate directions h_i along which to line minimize. After N such line minimizations, we would have efficiently reached the minimum of the quadratic equation.

For a better understanding on how this works, consider the Figure 3.4 below. This is in comparison to the steepest descent (Figure 3.3 shown above). Here, the number of iterations is only two where the second line is conjugate to the previous line traversed. This shows the efficacy of this method where we need 2 calls to reach the desired minimum value over a number of them used in steepest descent.

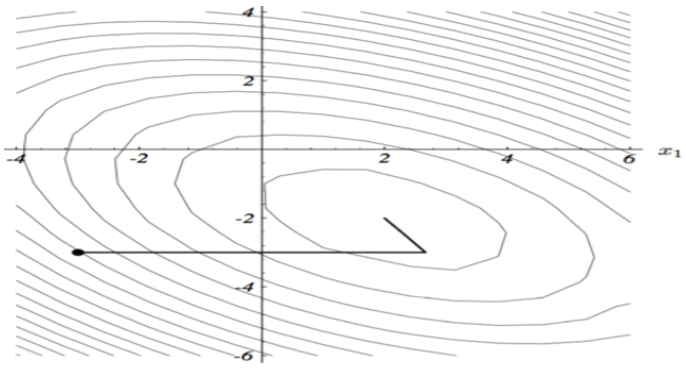

Figure 3.4: Conjugate Gradient Method.

The basic algorithm for the conjugate gradient method is as shown below. Here, the inputs are A, b, the starting value x, the maximum number of iterations i_{max} and an error tolerance ϵ ¡ 1. The initial iteration is set to zero. The algorithm terminates when i_{max} is exceeded. The fast recursive formula for the residual is used usually. But, it is exactly calculated once every 50 iterations (arbitrary). If n is large the number of iterations will be $\sqrt{n}$. The difference between this and the steepest descent algorithm

is that the residual 'r' is assigned to d. Hence, when a new path needs to be traversed, it is done in conjugate to this. This also ensures that no previous path traversed will be crossed.

$i \Leftarrow 0$
$r \Leftarrow b - Ax$
$d \Leftarrow r$
$\delta_{new} \Leftarrow r^T r$
$\delta_0 \Leftarrow \delta_{new}$
While $i < i_{max}$ **and** $\delta_{new} > \varepsilon^2 \delta_0$ **do**
    $q \Leftarrow Ad$
    $\alpha \Leftarrow \frac{\delta_{new}}{d^T q}$
    $x \Leftarrow x + \alpha d$
    If i **is divisible by 50**
        $r \Leftarrow b - Ax$
    else
        $r \Leftarrow r - \alpha q$
    $\delta_{old} \Leftarrow \delta_{new}$
    $\delta_{new} \Leftarrow r^T r$
    $\beta \Leftarrow \frac{\delta_{new}}{\delta_{old}}$
    $d \Leftarrow r + \beta d$
    $i \Leftarrow i + 1$

Figure 3.5: Conjugate Gradient Algorithm.

This is the conventional form of the conjugate gradient that is used to solve a system of linear equations. An in - built function is available in many programming softwares that essentially execute this algorithm. Since, we are dealing with a non - linear system, we use a non - linear conjugate gradient Fletcher - Reeve's method to update the Beta (β) values in the algorithm. The steps of the algorithm is as shown

below;

$$\text{Given } x_0;$$

$$\text{Evaluate } f_0 = f(x_0), \nabla f_0 = \nabla f(x_0);$$

$$\text{Set } p_0 \leftarrow -\nabla f_0, k \leftarrow 0;$$

$$\text{while } \nabla f_k \neq 0$$

$$\text{Compute } \alpha_k \text{ and set } x_{k+1} = x_k + \alpha_k p_k;$$

$$\text{Evaluate } \nabla f_{k+1};\ \beta_{k+1}^{FR} \leftarrow \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla {f_k}^T \nabla f_k};$$

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{FR} p_k;$$

$$k \leftarrow k + 1;$$

$$\text{end (while)}$$

Here, $f(x_0)$ will be computed by calling Robins code and then running the FFT routine to find A and B. ∇f_k will be calculated by running Robins code with changing the individual parameters one by one and calculating the partial derivative using the equation;

$$\nabla f_k = \frac{f(x+h) - f(x)}{h} \tag{3.9}$$

where h will be a very small percent change in the original value of parameter. α_k will be calculated by using back - track algorithm. We can also keep this constant as gradient will reduce at each step. After above code is run we will get new values of parameters based of which we will calculate new A, B and find the error and the repetition of the whole procedure follows again till we reach a threshold value on cost function.

Chapter 4

RESULTS

Once the issues were fixed and the challenges overcome, the implementation of the programs were done. This section discusses the results of the implementation of FFT and optimization algorithms. The FFT implementation shows a number of iterations. Each an improvement of the previous one.

4.1 Fast Fourier Transform Analysis

As mentioned previously, we use FFT for the accurate and efficient manipulation of our data. As we can see that our data has been collected over periodic intervals of time, we can say that our data is a discretely sampled data.

To use FFT for our data, we first need to run Robin's code. This will give us the values of the theoretical and experimental P_{CO_2} and P_{H_2O} at different frequencies. The frequency input generates a sinusoidal wave for P_{H_2O} and a corresponding sinusoidal wave for P_{CO_2}. The code is run at different time intervals and the output occurs in the form of sinusoidal waves. One such example is as shown in Figure 4.1 below. Here, we see the waves being generated for P_{H_2O} at 80 minute interval. This generates a corresponding P_{CO_2} wave as shown in Figure 4.2.

FFT is then applied to manipulate this graph and get the values of the amplitude. The FFT is applied to the theoretical and experimental data of P_{CO_2}. It is as shown in the figures 4.3 and 4.4 respectively. Since, we have only the theoretical data to work with and manipulate, we use only the first graph henceforth.

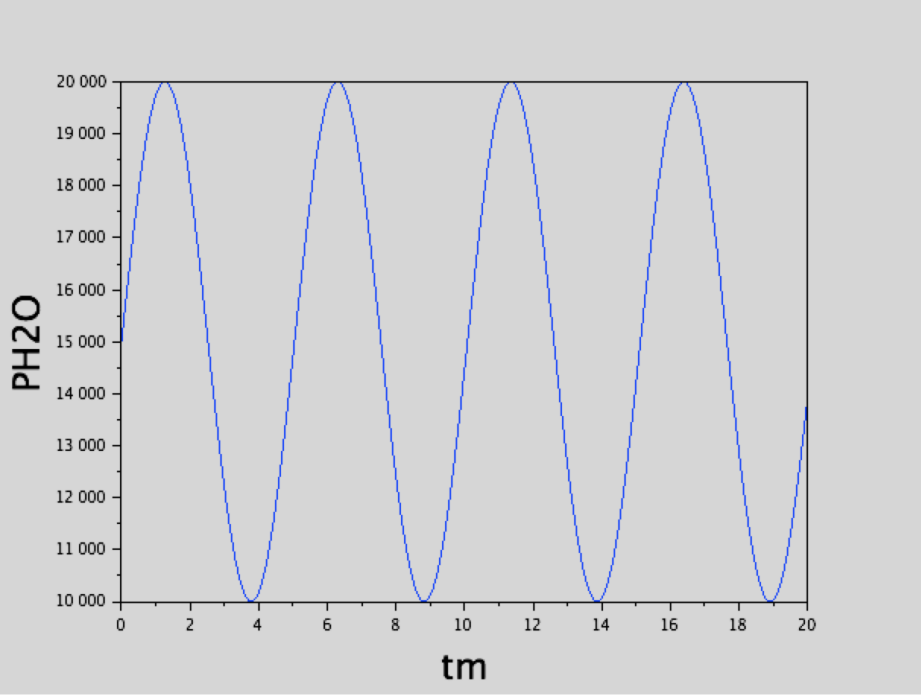

Figure 4.1: The Sinusoidal Wave of P_{H_2O} obtained for 80 min run.

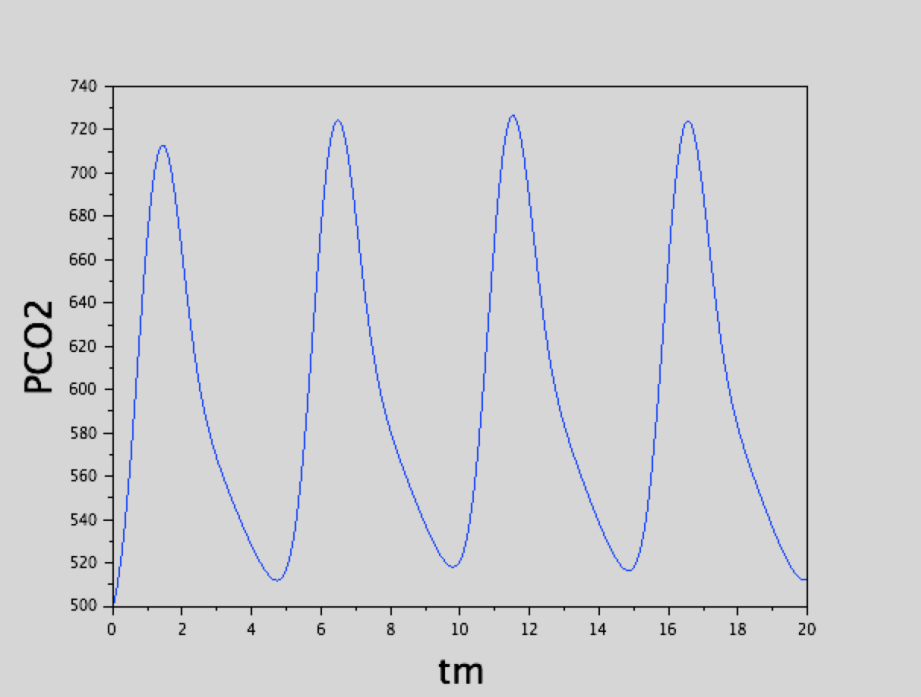

Figure 4.2: The Sinusoidal Wave of P_{CO_2} obtained for 80 min run.

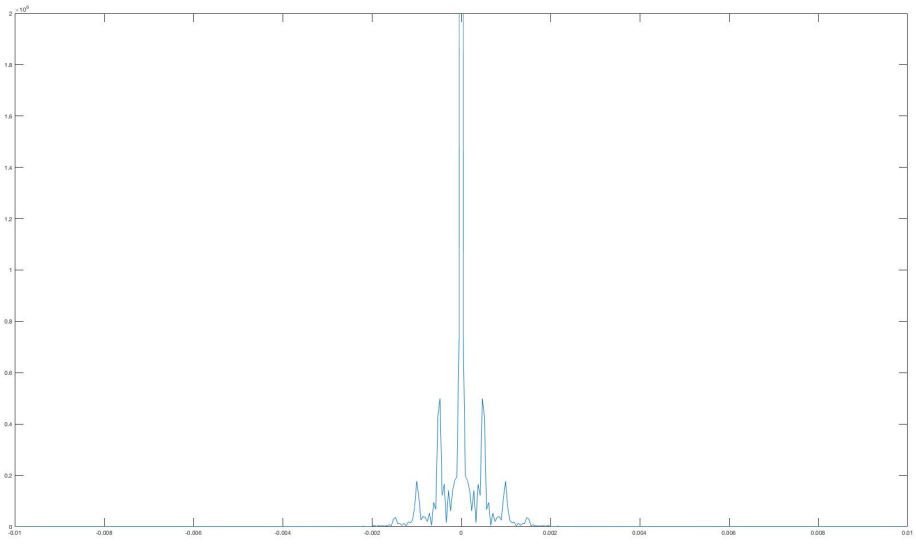

Figure 4.3: FFT of Theoretical P_{CO_2}

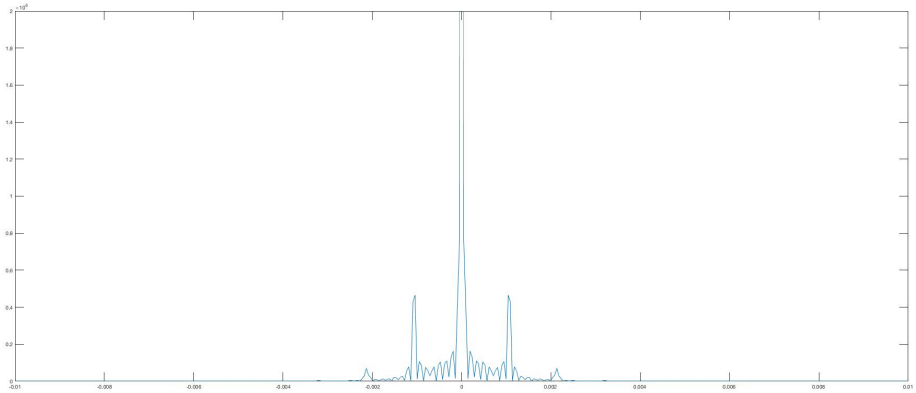

Figure 4.4: FFT of Experimental P_{CO_2}

As seen in the figure 4.3 above, the output of FFT has noise. Therefore, it can be smoothed using a Gaussian. Its zero frequency is also very large. This means that we have an average non-zero value. Also the width of the peaks seems to be about the same. Hence, it is essential to make it standard. This is done by using Gaussian by multiplying the signal with a Gaussian about one third the width of the time window. This will smear about the peaks in a well defined manner and we can then estimate

the strength at the two peaks at positive and negative frequencies as an integral over a narrow window. This can be shown in Figure 4.5 below;

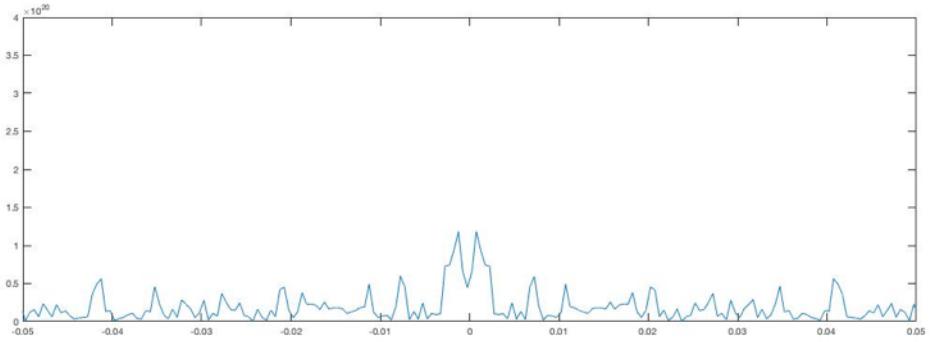

Figure 4.5: Peaks obtained from FFT after 1/3rd Gaussian is applied.

As we can see in this graph, the Fourier transform of a Gaussian is another Gaussian with the inverse width. It is the result of the convolution of the functions so that the peaks are smeared out in a well defined manner. However, in the graph above, we can see that the peaks of the small frequencies (noise) that should have been smeared out have actually been enhanced. This is because the Gaussian window that was used for this was very large. To overcome this, instead of convolving the two functions, I had to multiply it. I multiplied the signal with a function that is of the form;

$$\frac{1}{N} * \exp(\frac{-t}{t_0})^2 \tag{4.1}$$

Where the time interval runs from -T to T and $t_0 = T/3$. After implementing this change in the Gaussian window range and subtracting the zero mean value, I get the graph as shown in the figure 4.6 below.

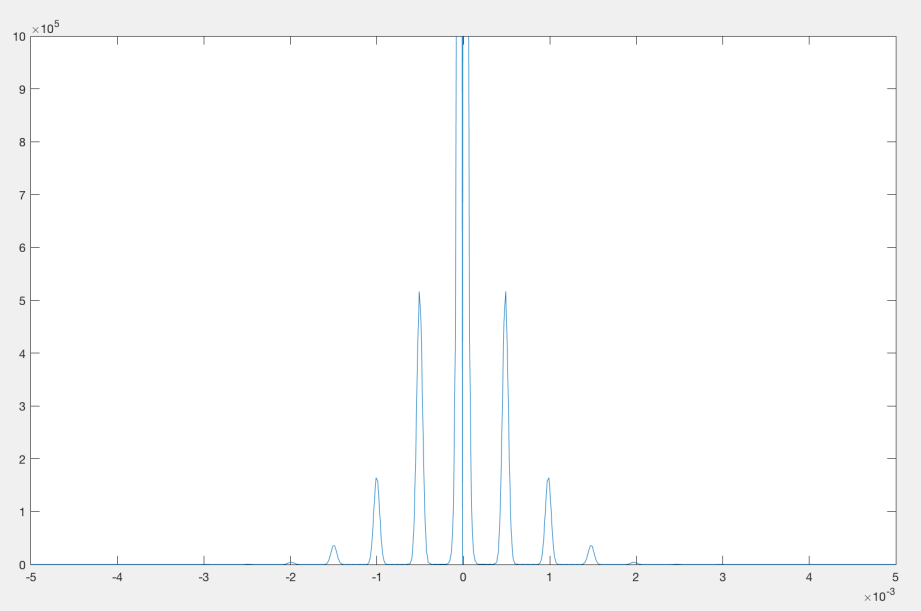

Figure 4.6: Peaks obtained from FFT after Gaussian Window is applied.

The FFT graph thus obtained is plotted in a complex frequency domain. Hence, the amplitudes obtained by the second peaks in this graph is a complex number on the negative and positive side. The value of 'A' is given by adding the real components of the values and the value of 'B' is given by adding the coefficient of imaginary components and removing 'i' by dividing it. Implementing this in the MATLAB code, we get the values of 'A' and 'B' as;

$$\text{A} = 1.2736\text{e+}07$$

$$\text{B} = 9.7841\text{e+}04$$

Now that we have the values of 'A' and 'B', the next step is to use this in the conjugate gradient algorithm along with the 8 parameters. We know that any A and B are related to the 8 parameters by some function 'f', that is, any change in any of the 8 parameters causes a change in the values of A and B. We do not know exactly how they are related but we can assume them to be related by some function 'f' as

defined below;

$$(A, B) = f(C_{OH^-}, C_{HCO^{-3}}, C_{CO_3^{2-}}, C_{H_2O}, K_1', k_a', k_p', p, \lambda) \tag{4.2}$$

4.2 Conjugate Gradient

As seen previously, the conventional form of conjugate gradient cannot be used. The conjugate gradient method widely used is to solve a linear system of equations. However, in our case, we have a non - linear system. Hence, we will be using it for a general minimization of the system given by the Fletcher - Reeve's method. Based on the algorithm explained in the previous chapter, a code was developed where the input is the 8 parameters as a function of A and B, the direction of minimization and the step size. We assume the initial values of the 8 parameters to be the experimental values. Then, we change all the values by a certain amount. We assume this to be the new theoretical values. This is as shown in the following table;

Table 4.1: Initial Values of the 8 Parameters

Initial (exp)	Changed (theor)
$C_{OH^-} = 4e-1$	4.1e-1
$C_{HCO_3{}^-} = 1.2e-1$	1.3e-1
$C_{CO_3{}^{2-}} = 1.5e-1$	1.6e-1
$K_1 = 4e+9$	4.1e+9
$K_a = 1e-5$	1e-5
$K_c = 4e-3$	4.1e-3
$K_p = 5e-12$	5.1e-12
p = 9.5	9.5

The corresponding A and B values for these are;

$$A_{exp} = 488520$$

$$B_{exp} = 5815.4$$

$$A_{th} = 489670$$

$$B_{th} = 5834.4$$

The next step is to change the values of each parameter. This is done to calculate the gradient that is an input in the optimization algorithm. Each run is done by changing one value of a parameter at a time and keeping the rest constant. Each run gives the corresponding A and B values. The following table shows all these values after 8 runs;

Table 4.2: Change in Values for Each Run and the Corresponding A and B values

C_{OH^-}	4.2e-1							
$C_{HCO_3^-}$		1.4e-1						
$C_{CO_3^{2-}}$			1.7e-1					
K_1				4.2e+9				
K_a					1e-5			
K_c						4.2e-3		
K_p							5.2e-12	
p								9.5
A	489260	486400	493210	488450	489670	490930	490820	489670
B	5801.8	5905.9	5737.8	5715.8	5834.4	5956.1	5943.2	5834.4

As we can see from the above tables, I have kept the values of K_a and p as the same. This is because we saw an erratic behavior in the model when these values were changed. The effect of change in values of K_a and p are shown in the figures below;

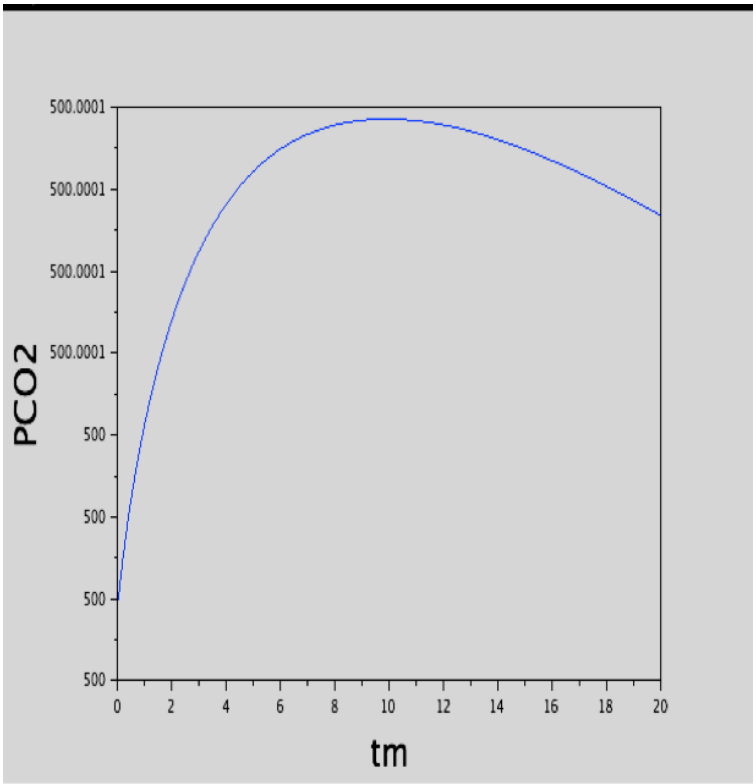

Figure 4.7: P_{CO_2} Graph from Small Change in K_a.

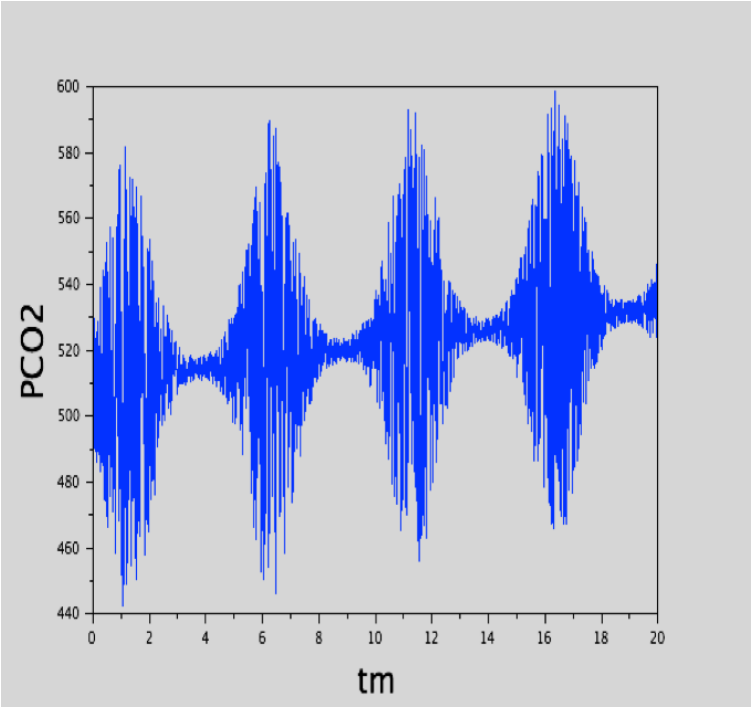

Figure 4.8: P_{CO_2} Graph from Small Change in p.

Hence, using these 8 parameter values, their corresponding gradient and a step size of 0.5, we run the conjugate gradient code. The following parameter values is a result of the minimization in our code.

Table 4.3: Output of the Minimized Values

C_{OH^-}	-3.65e+1							
$C_{HCO_3^-}$		-3.27e+2						
$C_{CO_3^{2-}}$			-5.86e+2					
K_1				4.1e+9				
K_a					1e-5			
K_c						-1.18e+3		
K_p							-9.3e+12	
p								9.5

When these values were used in our numerical simulation, we saw that the code wouldn't run and give a negative P_{CO_2} graph. This is possibly because of a larger difference in our assumed values. This can be rectified by assuming a smaller difference or by using actual experimental values. This whole process constitutes a single iteration. Multiple iterations must be executed till we reach a stage where the error is the least.

Chapter 5

CONCLUSION AND FUTURE WORK

In order to validate Isabelle's theory, it was essential to simulate the model numerically and match it with an identical experimental study. Robin's thesis proves this. However, the study was based on a number of assumed parameter values. He proved the behavior of the resin. The next step was to optimize these parameters that govern the entire mechanism. This is done by using the conjugate gradient method. Prior to optimization, the range of potential values were gathered using the Fast Fourier Transform on different input frequencies in Robin's code to study the different effects it has on the values of P_{H_2O} and P_{CO_2}.

The future work involves a number of improvements that need to be done in order to get the most accurate results. From the experimentation side, it is important to redo the experiments and get more data to work with. Right now, we have very limited amount of data experimentally to substantiate our claims. On the numerical simulation side, in order to get accurate optimization results, the model needs to be solved for non-linear. Presently, we are linearizing to get find the nearest approximation to provide the best estimation for the fitting parameters.

REFERENCES

Abs, R., "Modelization and calibration of moisture sensitive beads in carbon dioxide capture.", (2014).

Change, C., "Ipcc fourth assessment report", The Physical Science Basis **2**, 580–595 (2007).

Press, W. H., *Numerical recipes 3rd edition: The art of scientific computing* (Cambridge university press, 2007).

Remy, I., "Numerical simulation of carbon dioxide pump", (2014).

Shewchuk, J. R., "An introduction to the conjugate gradient method without the agonizing pain", (1994).

Wang, T., K. S. Lackner and A. Wright, "Moisture swing sorbent for carbon dioxide capture from ambient air", Environmental science & technology **45**, 15, 6670–6675 (2011).